

PATENT ABSTRACTS OF JAPAN

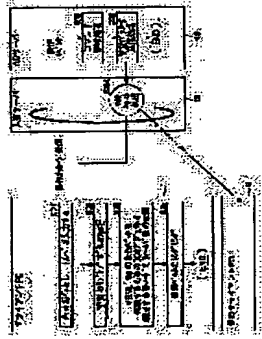
(11)Publication number: 11-212934  
(43)Date of publication of application : 06.08.1999

(51)Int.Cl.	G06F 15/16
	A63F 9/22
	G06F 13/00
	G06T 15/00
	G09G 5/00

(21)Application number: 10-026677 (71)Applicant : SONY CORP  
(22)Date of filing : 23.01.1998 (72)Inventor : MATSUDA KOICHI  
NAITOU TAKEHITO  
UENO HIROSHI

(54) INFORMATION PROCESSING DEVICE AND METHOD AND INFORMATION SUPPLY MEDIUM

(57)Abstract:  
PROBLEM TO BE SOLVED: To make a virtual pet perform tricks.  
SOLUTION: A user serving as the owner of a virtual pet inputs a command via a chat operation in a step S21, and this command is sent to an AO server 13. The server 13 holds a table where the commands and their corresponding actions of the pet are described, for instance, the corresponding relation is stored between a command 'Jump !' and its corresponding action 'jumping'. Thus, the server 13 analyzes every received command based on the table. Then the analysis results of commands are sent to a client PC 1 in a step S23. The client PC 1 executes the processes defined in a step S25 and its following ones based on the received analysis results of commands. Thus, the pet performs tricks such as the dancing, etc.



BEST AVAILABLE COPY

(12) 公開特許公報 (A)

(19) 日本国特許庁 (JP)

(11) 特許出願公開番号

特開平 11-212934  
(43) 公開日 平成 11 年 (1999) 8 月 6 日

(51) Int. Cl. *	分類記号	F I
G 06 F 15/16	4 3 0	G 06 F 15/16 4 3 0 B
A 63 F 9/22		A 63 F 9/22 C
G 06 F 13/00	3 5 1	G 06 F 13/00 3 5 1 E
G 06 T 15/00		G 09 G 5/00
G 09 G 5/00		G 06 F 15/62 3 6 0
審査請求 未請求	請求項の数 7	F D (全 35 頁)

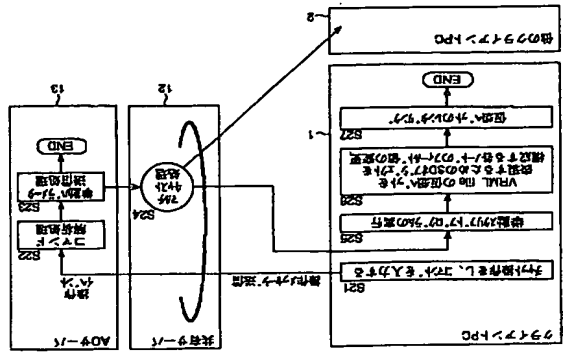
(21) 出願番号	特願平 10-26677	(71) 出願人	000002185 ソニー株式会社 東京都品川区北品川 6 丁目 7 番 35 号
(22) 出願日	平成 10 年 (1998) 1 月 23 日	(72) 発明者	松田 晃一 東京都品川区北品川 6 丁目 7 番 35 号 株式会社内 内藤 剛人 東京都品川区北品川 6 丁目 7 番 35 号 株式会社内 上野 比呂至 東京都品川区北品川 6 丁目 7 番 35 号 株式会社内 井理士 稲本 義雄

特許法第 64 条第 2 項ただし書の規定により図面第 12 図、17 図、18 図、20 図、21 図、23 図の一部は不掲載とした。

(54) 【発明の名称】 情報処理装置および方法、並びに提供媒体

(57) 【要約】

【課題】 仮想ペットに芸をさせる。  
【解決手段】 仮想ペットの飼い主であるユーザは、ステップ S 21 において、チャット操作によりコマンドを入力する。このコマンドは AO サーバ 13 に送信される。AO サーバ 13 は、コマンドとそのコマンドに対応する仮想ペットが行う動作について書かれたテーブル、例えば、コマンド "跳" に対しての動作 "その場で跳ねる" といった対応関係が記憶されたテーブルを保持し、そのテーブルに基づいて送信されたコマンドを解析する。その解析結果は、ステップ S 23 において、クライアント PC 1 に送信される。クライアント PC 1 においては、受信された解析結果を基に、ステップ S 25 以下の処理が行われ、仮想ペットがダンスなどの芸を行う。



【特許請求の範囲】

【請求項1】 他の情報処理装置とともに、ネットワークを介してサーバーに接続され、共有仮想空間の提供を受ける情報処理装置において、

前記共有仮想空間において飼育される仮想生命オブジェクトに対して、所定の動作をさせるコマンドを入力する入力手段と、

前記コマンドを前記サーバーに送信する送信手段と、前記サーバーから伝送される前記コマンドに対する前記仮想生命オブジェクトの挙動を制御するスク립トを受信する受信手段と、

前記スク립トに基づいて、前記仮想生命オブジェクトの表示を制御する表示制御手段とを備えることを特徴とする情報処理装置。

【請求項2】 前記コマンドは、漢字と記号の組み合わせであることを特徴とする請求項1に記載の情報処理装置。

【請求項3】 他の情報処理装置とともに、ネットワークを介してサーバーに接続され、共有仮想空間の提供を受ける情報処理装置の情報処理方法において、

前記共有仮想空間において飼育される仮想生命オブジェクトに対して、所定の動作をさせるコマンドを入力する入力ステップと、

前記コマンドを前記サーバーに送信する送信ステップと、前記サーバーから伝送される前記コマンドに対する前記仮想生命オブジェクトの挙動を制御するスク립トを受信する受信ステップと、

前記スク립トに基づいて、前記仮想生命オブジェクトの表示を制御する表示制御ステップとを備えることを特徴とする情報処理方法。

【請求項4】 他の情報処理装置とともに、ネットワークを介してサーバーに接続され、共有仮想空間の提供を受ける情報処理装置に用いるコンピュータプログラムにおいて、

前記共有仮想空間において飼育される仮想生命オブジェクトに対して、所定の動作をさせるコマンドを入力する入力ステップと、

前記コマンドを前記サーバーに送信する送信ステップと、前記サーバーから伝送される前記コマンドに対する前記仮想生命オブジェクトの挙動を制御するスク립トを受信する受信ステップと、

前記スク립トに基づいて、前記仮想生命オブジェクトの表示を制御する表示制御ステップとを備えることを特徴とするコンピュータプログラム。

【請求項5】 ネットワークを介して接続されている複数のクライアント装置に共有仮想空間を提供する情報処理装置において、

前記共有仮想空間において飼育される仮想生命オブジェクトが行う所定の動作に関するコマンドを前記クライアント装置から受信する受信手段と、

【請求項6】 前記クライアント装置に共有仮想空間を提供する情報処理装置において、

前記共有仮想空間において飼育される仮想生命オブジェクトが行う所定の動作に関するコマンドを前記クライアント装置から受信する受信手段と、

【請求項7】 ネットワークを介して接続されている複数のクライアント装置に共有仮想空間を提供する情報処理装置において、

前記共有仮想空間において飼育される仮想生命オブジェクトが行う所定の動作に関するコマンドを前記クライアント装置から受信する受信手段と、

【請求項8】 ネットワークを介して接続されている複数のクライアント装置に共有仮想空間を提供する情報処理装置において、

介してセンターのホストコンピュータに接続し、所定の通信プロトコルに基づいてホストコンピュータにアクセスする、いわゆるパソコン通信サービスの分野においては、Habitat（商標）と呼ばれるサイバースペースのサービスが知られている。

【0003】Habitatは、米国LucasFilm社によって1985年から開発が始められ、米国商業ネットワークであるQuintumLink社で約3年間運用された後、1990年2月に富士通Habitat（商標）としてNIFTY-Serveでそのサービスが開始されたものである。このHabitatにおいては、2次元グラフィックスで描画された「ポピュロポリス（Populopolis）」と呼ばれる仮想の都市に、アバタ（avatar；イン）と呼ばれる仮想の化身）と呼ばれるユーザの分身を送り込み、ユーザ同士がチャット（Chat；文字の入力と表示によるテキストベースでのリアルタイムの対話）などを行うことができる。このHabitatの更に詳細な説明については、サイバースペース、マイケル・ベネディクト編、1994年9月20日初版発行、N.T.T出版 ISBN 4-87188-265-900010（原著：Cyberspace：First Steps Michael Benedikt, ed. 1991, MIT Press/Cambridge, MA ISBN0-262-02327-X）第282頁乃至第307頁を参照されたい。

【0004】この種のパソコン通信サービスで運用されている従来のサイバースペースシステムにおいては、仮想的な街並みや部屋の内部の様子を3次元グラフィックスで描画されており、アバタを操作するまたは手前方向へ移動させる場合、単にアバタを3次元グラフィックスの背景上や上下に移動させるだけであり、仮想空間内での歩行や移動を疑似体験させるには表示の上での表現力が乏しかった。また、自分の分身であるアバタと他人のアバタが表示された仮想空間を、第3者の視点で見ることになるため、この点においても、疑似体験の感覚が損なわれるものであった。

【0005】そこで、特開平9-81781号公報に開示されているように、仮想空間を3次元グラフィックスで表示し、ユーザがアバタの視点で自由に歩き回れる機能が、VRML（Virtual Reality Modeling Language）と呼ばれる3次元グラフィックス・データの記述言語を利用することによって実現されている。また、ユーザの代理となるアバタを用いてチャットを行う種々のサイバースペースの考案に関しては、日経エレクトロニクス1996.9.9（no. 670）の第151頁乃至159頁に記載されている。

【0006】一方、近年、パーソナルコンピュータ用のソフトウェア・プログラムとして、熱帯魚を育てる飼育シミュレーション・ゲームや、仮想世界に住んでいる人々の知識をもった仮想生物を育てる飼育シミュレーション・ゲームなどが知られている。また、電子手帳に犬や猫等の疑似ペットを表示、その成長過程を楽しむようにした製品も知られている（日経エレクトロニクス1997.4.7（no. 686）の第131頁乃至134頁参照）。さらに、この

【0007】この種の仮想電子ペットは、1チップのLSI（大規模集積回路）にCPU（中央処理装置）やROM、RAM等が内蔵されており、そのROM内に飼育シミュレーション・ゲーム・プログラムが格納され、さらにLCD（液晶表示装置）によって仮想的なペットの姿や状態が表示されるようになっている。ユーザは、操作ボタンを操作して、「食事を与える」、「掃除をする」など、ペットとしての仮想生物を飼育するのに必要な指示を与える。この結果として、LCDで表示される仮想生物が成長し、その成長の過程で、仮想生物の外観が、例えば、たまご、ひよこ、成長へと段階的に変化していく。

【0008】また、ユーザのボタン操作に応じて、その指示が適切ならば順調に仮想生物が成長し、不適切ならば、病気になったり死亡してしまうようにプログラムされている。さらに、内部に組み込まれているカレンダー・タイマーによって得られる仮想生物の誕生時点からの経過時間に基づいて、仮想生物側より各種の要求が行われるようにプログラムされており、例えば夜間の時間帯では仮想生物より睡眠の要求がなされ、食事の時間帯では食事の要求があり、またランダムにおやつや遊びの要求がなされる。これらの要求にユーザが適切に対応しない場合、仮想生物の成長が遅れたり、性格が悪化したりする。一方、ユーザが適切に対応した場合には、仮想生物の寿命が伸びるようプログラムされている。

【0009】ここで、例えば、特開平07-160853号公報には、電子手帳などに適用され、動物や植物等の生物の成長過程に応じた画像を表示する技術が開示されている。すなわち、植物やキャラクターの成長過程の各段階のどのマッピング画像をROMに格納しておき、成長度に応じた植物やキャラクターをLCDに表示させると共に、予めROMに記憶されている植物成長要素（水、光、肥料）の各キャラクターを表示させ、それら成長要素の投入量をキー入力することで、その投入量に応じた各成長要素の値がROM内の水レベルレジスタ、光レベルレジスタ、肥料レベルレジスタに各々セットされ、これらの各レベルレジスタの値に基づいて、新たな成長度が算出され、その算出された成長度に対応した植物やキャラクターがROMから読み出されてLCDに表示される。これにより、ユーザの飼育状況に応じた植物の成長過程が表示される。

【0010】

【発明が解決しようとする課題】しかしながら、上述したような仮想空間で仮想ペットに所定の動作をさせる場合、ユーザに、予め用意されているボタンを選択させるようにしていた。その結果、仮想ペットに複雑な動作や、独自の動作をさせることができない課題があった。

【0011】本発明はこのような状況に鑑みてなされた

の種の飼育シミュレーション・ゲーム・プログラムを組込んだ。たまごの携帯電子ペットとして、バンダイ株式会社が開発・製品化した「たまごっち（商標）」が広く知られている。

【0007】この種の仮想電子ペットは、1チップのLSI（大規模集積回路）にCPU（中央処理装置）やROM、RAM等が内蔵されており、そのROM内に飼育シミュレーション・ゲーム・プログラムが格納され、さらにLCD（液晶表示装置）によって仮想的なペットの姿や状態が表示されるようになっている。ユーザは、操作ボタンを操作して、「食事を与える」、「掃除をする」など、ペットとしての仮想生物を飼育するのに必要な指示を与える。この結果として、LCDで表示される仮想生物が成長し、その成長の過程で、仮想生物の外観が、例えば、たまご、ひよこ、成長へと段階的に変化していく。

【0008】また、ユーザのボタン操作に応じて、その指示が適切ならば順調に仮想生物が成長し、不適切ならば、病気になったり死亡してしまうようにプログラムされている。さらに、内部に組み込まれているカレンダー・タイマーによって得られる仮想生物の誕生時点からの経過時間に基づいて、仮想生物側より各種の要求が行われるようにプログラムされており、例えば夜間の時間帯では仮想生物より睡眠の要求がなされ、食事の時間帯では食事の要求があり、またランダムにおやつや遊びの要求がなされる。これらの要求にユーザが適切に対応しない場合、仮想生物の成長が遅れたり、性格が悪化したりする。一方、ユーザが適切に対応した場合には、仮想生物の寿命が伸びるようプログラムされている。

【0009】ここで、例えば、特開平07-160853号公報には、電子手帳などに適用され、動物や植物等の生物の成長過程に応じた画像を表示する技術が開示されている。すなわち、植物やキャラクターの成長過程の各段階のどのマッピング画像をROMに格納しておき、成長度に応じた植物やキャラクターをLCDに表示させると共に、予めROMに記憶されている植物成長要素（水、光、肥料）の各キャラクターを表示させ、それら成長要素の投入量をキー入力することで、その投入量に応じた各成長要素の値がROM内の水レベルレジスタ、光レベルレジスタ、肥料レベルレジスタに各々セットされ、これらの各レベルレジスタの値に基づいて、新たな成長度が算出され、その算出された成長度に対応した植物やキャラクターがROMから読み出されてLCDに表示される。これにより、ユーザの飼育状況に応じた植物の成長過程が表示される。

【0010】

【発明が解決しようとする課題】しかしながら、上述したような仮想空間で仮想ペットに所定の動作をさせる場合、ユーザに、予め用意されているボタンを選択させるようにしていた。その結果、仮想ペットに複雑な動作や、独自の動作をさせることができない課題があった。

【0011】本発明はこのような状況に鑑みてなされた

【0012】

【0013】

【0014】

【0015】

【0016】

【0017】

【0018】

【0019】

【0020】

【0021】

【0022】

送する伝送ステップとを備えることを特徴とする。

【0017】請求項7に記載の提供媒体は、ユーザが共有仮想空間において飼育される仮想生命オブジェクトが行う所定の動作に関するコマンドをクライアント装置から受信する受信ステップと、コマンドと仮想生命オブジェクトが行う所定の動作とに対応関係が記述されたデータベースを保持する保持ステップと、受信ステップで受信されたコマンドを、データベースを参照し、解析する解析ステップと、解析ステップの解析結果に基づいて仮想生命オブジェクトの表示を制御するスク립トをクライアント装置に伝送する伝送ステップとを備えるコンピュータプログラムを提供することを特徴とする。

【0018】請求項1に記載の情報処理装置、請求項3に記載の情報処理方法、および請求項4に記載の提供媒体においては、共有仮想空間において飼育される仮想生命オブジェクトに対して、所定の動作をさせるコマンドが入力され、入力されたコマンドがサーバに送信され、サーバから送信されたコマンドに対する仮想生命オブジェクトの挙動を制御するスク립トが受信され、このスク립トに基づいて、仮想生命オブジェクトの表示が制御される。

【0019】請求項5に記載の情報処理装置、請求項6に記載の情報処理方法、および請求項7に記載の提供媒体においては、共有仮想空間において飼育される仮想生命オブジェクトが行う所定の動作に関するコマンドが受信され、コマンドと仮想生命オブジェクトが行う所定の動作との関係が記述されたデータベースが保持され、受信されたコマンドを、データベースを参照し、解析し、その解析結果に基づいて仮想生命オブジェクトの表示を制御するスク립トがクライアント装置に伝送される。

【0020】  
【発明の実施の形態】以下に本発明の実施の形態を説明するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し一例）を付加して本発明の特徴を記述すると、次のようになる。但し、勿論この記載は、各手段を記載したものに限定することを意味するものではない。

【0021】請求項1に記載の情報処理装置は、共有仮想空間において飼育される仮想生命オブジェクトに対して、所定の動作をさせるコマンドが入力される入力手段（例えば、図2のステップS22）と、入力手段により入力されたコマンドをサーバに送信する送信手段（例えば、図4のモジュール39）と、サーバから伝送されたコマンドに対する仮想生命オブジェクトの挙動を制御するスク립トを受信する受信手段（例えば、図4のモジュール39）と、スク립トに基づいて、仮想生命オブジェクトの表示を制御する表示制御手段（例えば、図2のステップS26）とを備えることを特徴とする。

【0022】請求項5に記載の情報処理装置は、共有仮

仮想空間において飼育される仮想生命オブジェクトが行う所定の動作に関するコマンドをクライアント装置から受信する受信手段（例えば、図7のAOサーバ13）と、コマンドと仮想生命オブジェクトが行う所定の動作との対応関係が記述されたデータベース（例えば、図7の舉動コマンド管理テーブル）を保持する保持手段（例えば、図7のHDD13a）と、受信手段において受信されたコマンドを、データベースを参照し、解析する解析手段（例えば、図2のステップS22）と、解析手段の解析結果に基づいて仮想生命オブジェクトの表示を制御するスク립トをクライアント装置に伝送する伝送手段（例えば、図2のステップS23）とを備えることを特徴とする。

【0023】以下、図面を参照し、本発明の実施の形態について説明する。

【0024】説明に先立ち、世界的規模で接続されたコンピュータネットワークであるインターネット(The Internet)において、様々な情報を提供するWWW(world wide web)の枠組みを利用して3次元的な情報を統一的に扱うことができる記述言語であるVRML(virtual reality modeling language)について説明する。

【0025】ここで、インターネットで利用できる情報提供システムとして、スイスのCERN(European Center for Nuclear Research:欧州核物理学研究所)が開発したWWWが知られている。これは、テキスト、画像、音声などの情報をハイパーテキスト形式で閲覧できるようにしたもので、HTTP(Hyper Text Transfer Protocol)と呼ばれるプロトコルに基づいて、WWWサーバに格納された情報をパーソナルコンピュータなどの端末に非同期伝送するものである。

【0026】WWWサーバは、HTTPデーモン(HTTP:Hyper Text Transfer Protocol、ハイパーテキスト転送プロトコル)と呼ばれるサーバ・ソフトウェアとハイパーテキスト情報が格納されたHTMLファイルによって構成される。なお、デーモンとは、UNIX上で作業を行う場合、バックグラウンドで管理や処理を実行するプログラムを意味する。ハイパーテキスト情報は、HTML(HyperText Markup Language、ハイパーテキスト記述言語)と呼ばれる記述言語によって表現される。HTMLによるハイパーテキストの記述は、「<」と「>」で囲まれたタグと呼ばれる書式指定によって文章の論理的な構造が表現される。他の情報とのリンクについての記述は、アンカーと呼ばれるリンク情報によって行われる。アンカーによって情報が存在する所を指定する際には、URL(Uniform Resource Locator)が用いられる。

【0027】HTMLで記述されたファイルは、TCP/IP(Tra nsmission Control Protocol/Internet Protocol)ネットワーク上において転送するためのプロトコルがHTTPである。クライアントからの情報の要求をWWWサーバに伝え、HTMLファイルのハイパーテキスト情報をクライアント

トに転送する機能をもっている。  
【0028】WWWを利用する環境として多く利用されているのが、WWWブラウザ(ブラウザとは閲覧するという意味)と呼ばれる、Netscape Navigator(米国Netscape Communications社の商標)をはじめとするクライアント・ソフトウェアである。

【0029】このWWWブラウザを用いて、URLに対応する世界規模で広がるインターネット上のWWWサーバ上のファイル、いわゆるホームページを閲覧することができる。いわゆるネットサーフィンと呼ばれるように、リンクが張られたホームページを次々と辿って、多種多様なWWWの情報ソースにアクセスすることができる。

【0030】近年、このWWWをさらに拡張し、3次元空間の記述や、3次元グラフィックスで描画されたオブジェクトに対してハイパーテキストのリンクの設定を可能とし、これらのリンクをたどりながらWWWサーバを次々とアクセスできるようにしたVRMLと呼ばれる3次元グラフィックス記述言語で記述された3次元空間を提示するWWWブラウザが開発されている。

【0031】このVRMLの詳細は、例えば、「VRMLを知る:3次元仮想空間の構築とブラウザング」[マーク・ベッジン著、松田晃一・薄地真由・竹内彰一・本田麻生・暦本純一・石川真之・宮下健一・原和弘訳、1996年3月25日初版発行、プレジデントホール出版ISBN4-931356-37-0](原著:VRML:Browsing & Building Cyberspace,Mark Pesce,1995 New Readers Publishing ISBN 1-56205-498-8)J、および「VRMLの最新動向とCyber Passage」(松田晃一・本田晃生著、bit(共出版版)/1996 Vol.28 No.7 pp29乃至pp36, No.8 pp57乃至pp65, No.9pp29乃至pp36, No.10 pp49乃至pp58)J等の文献に記載されている。

【0032】また、August 4, 1996におけるThe Virtual Reality Modeling Language Version 2.0, ISO/IEC CD 14772の公式かつ完全な仕様書は、http://webpac.e.sgi.com/moving-worlds/spec/index.htmlで公開されており、その日本語版は、http://www.webcity.co.jp/info/andoh/VRML/vrml2.0/spec-jp/index.htmlで公開されている。

【0033】さらに、VRML 2.0用ブラウザおよび共有サーバ用ソフトウェアとしては、例えば、本国人であるソニー株式会社「Community Place Browser/Bureau(商標)」として開発し、製品化したホームページhttp://www.sony.co.jpからダウンロード可能としている。

【0034】このようなVRML 2.0を用いた3次元な仮想空間を構築しようとする場合、まず、VRMLにより仮想空間内の物体(モデル)の形、動きおよび位置等を示す図形データの作成(モデル作成)、ユーザが画面表示された仮想空間内のモデルを、例えば、マウスでクリックしてポインタリングした場合にイベントを発生させる

スイッチ (センサ) のモデルへの付加 (センサ付加)、センサへのポインティングに応じて発生するイベントを実現するスクリプトのプログラミング (スクリプト作成)、センサに対する操作とスクリプトの起動等、図形データおよびスクリプト (以下、図形データ、スクリプトおよびVRMLに規定されているライク等の共通ノード等を総称してノードとも記す) の間の対応付け (ルーティン等) などによって所望のコンテンツを表現するVRMLファイルを作成する。

【0035】例えば、<http://www.ses.co.jp/SES/STAFF10/kan/howto/howto1.html> には、VRML2.0の書き方、サンプルデータなど、丁寧に解説されている。その一部を紹介すると、以下の通りである。

【0036】1. VRML2.0で書かれたワールドを見るにはHTML形式で書かれたデータを見るためにHTMLブラウザが必要となる。VRML2.0形式で書かれたデータを見るためにはVRML2.0に対応したVRMLブラウザが必要となる。尚、このページで作成したVRML2.0のデータは全てSONY社製のCommunity Place Browserで行なわれている。また、ファイルがVRMLで書かれたことを表すためにファイルの拡張子を\*.wrl (ワールドの意) に、更にVRML2.0で書かれたことを表すためにファイルの1行目に#VRML V2.0 utf8

と書くことが必要である。

【0037】2. VRML2.0データの基本構造  
VRML2.0のデータはノード(Node)とフィールド(Field)で構成されていて、基本的に次のような形で書かれている。

Node ( Field(s) )  
この中でFieldsは省略することができるが、Nodeと中かっこ(「」)は省略することができない。フィールドはノードに属性を渡し、ノードのパラメータを指定します。フィールドを省略した場合、デフォルト値が用いられる。また、フィールドには単一の値しか持たない(単一値フィールド(SF))と複数の値を持つ(複値フィールド

Sample2

```
1: #VRML V2.0 utf8
2: Transform (
3:   children (
4:     Shape (
5:       appearance Appearance (
6:         material Material ( diffuseColor 1 0 0 )
7:       )
8:       geometry Sphere ( radius 1 )
9:     )
10:   )
11: )
```

【0042】この球は赤色で表示される。追加されたのは5行目から7行目までである。diffuseColorフィールドのデータタイプは、SFColorであるから、RGBカラーを数

ド(MF)があり、単一値フィールドの名前は「SF」で、複値フィールドは「MF」で始まるようになっている。

【0038】3. 球の書き方  
VRML2.0では球や立方体、円柱、円錐などの基本図形を描くためのノードが用意されている。前にも書いているが、それぞれのノードはパラメータを指定するためのフィールドを持っており、次のように書くことで半径1の球を表示する。尚、radiusフィールドのデータタイプは、SFfloatです。1つの浮動小数点数の値をとる。

【0039】

```
Sample1
1: #VRML V2.0 utf8
2: Transform (
3:   children (
4:     Shape (
5:       geometry Sphere ( radius 1 )
6:     )
7:   )
8: )
```

【0040】実は2,3行目、7,8行目は書くなくてもかまいません。1行目と4行目乃至6行目だけで、きちんと半径1の球を表示できる。Transformノードは、Groupノードという種類のノードの1つで、文字通りノードのグループ化を行うためのノードである。Transformノードを含めて、後でく他のノードの詳しい機能やフィールドなどは「Appendix1: VRML2.0 Node List」を参照されたい。球を書くためのノードはSphereであるが、このノードはGeometryノードと呼ばれるノードの1つである。Geometryノードは、見え方と形状を定義するShapeノードのgeometryフィールドに書く必要がある。

【0041】4. 球に色を付ける

先ほどの球に色を付けるには次のように書く。

appearanceノードはShapeノードのappearanceフィールドに書くことになっている。このためこのような一見複雑な構造になっている。

【0043】5. テクスチャの張り付け \*  
ここでは予め用意したGIFイメージを使っている。

Sample3

```
1: #VRML V2.0 utf8
2: Transform (
3:   children (
4:     Shape (
5:       appearance Appearance (
6:         texture ImageTexture ( url "image.gif" )
7:       )
8:       geometry Box (
9:       )
10:     )
11:   )
```

【0044】テクスチャを張り付けているのは6行目で、※のノードである。

ある。ImageTextureノードはAppearanceノードのtextureフィールドに書くことになっているため、このような書き方になる。尚、8行目のBoxノードは立方体を描くため※20

※のノードである。

【0045】6. 物体の位置を移動させる

次にこの赤色の球を右に移動させる。

Sample4

```
1: #VRML V2.0 utf8
2: Transform (
3:   translation 2 0 0
4:   children (
5:     Shape (
6:       appearance Appearance (
7:         material Material ( diffuseColor 1 0 0 )
8:       )
9:       geometry Sphere ( radius 1 )
10:     )
11:   )
12: )
```

【0046】3行目に追加したtranslationフィールドで、物体を平行移動させる。translationフィールドは、translation x y z

は右方向に2だけ平行移動することになる。

で、それぞれx軸、y軸、z軸の移動量を指定する。ブラ

ウザ上ではx軸は左右 (右方向に+)、y軸は上下 (上方

向に+)、z軸は奥行き (手前に向かって+) を表す。従★

向に+)、z軸は奥行き (手前に向かって+) を表す。従★

Sample5

```
1: #VRML V2.0 utf8
2: Transform (
3:   translation 2 0 0
4:   children (
5:     Shape (
6:       appearance Appearance (
7:         material Material ( diffuseColor 1 0 0 )
8:       )
9:       geometry Sphere ( radius 1 )
10:     )
```

10: )

14

13

```

11: ]
12: }
13: Transform {
14:   translation -2 0 0
15:   children [
16:     Shape {
17:       appearance Appearance {
18:         material Material { diffuseColor 0 1 0 }
19:       }
20:       geometry Cylinder {}
21:     }
22:   ]
23: }

```

【0048】13行目以降に追加したソースは12行目以前のものと構造は同じである。物体が球から円錐へ、色が赤から緑へ、そして位置が左に移動している点が12行目以前のものと異なる。

【0049】8. ポリゴンで書く  
 上述した「VRML2.0の書き方」ではポリミティブな図形を使ったが、複雑な図形を書くときにはポリゴンを使用する。ポリゴンで表示するためにIndexedLineSet、IndexedFaceSetの2つのノードが用意されている。IndexedLineSetは線を、IndexedFaceSetは面を表現する。

15

(9)

16

```

1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     Shape {
5:       appearance Appearance {
6:         material Material { diffuseColor 0 0 1 }
7:       }
8:       geometry Sphere { radius 2 }
9:     }
10:   ]
11: }
12: Transform {
13:   translation 0 5 0
14:   children [
15:     Shape {
16:       appearance Appearance {
17:         material Material { diffuseColor 0 0 1 }
18:       }
19:       geometry Sphere { radius 2 }
20:     }
21:   ]
22: }

```

Sample7

```

1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     DEF BlueSphere Shape {
5:       appearance Appearance {
6:         material Material { diffuseColor 0 0 1 }
7:       }
8:       geometry Sphere { radius 2 }
9:     }
10:   ]
11: }
12: Transform {
13:   translation 0 5 0
14:   children [
15:     Shape {
16:       appearance Appearance {
17:         material Material { diffuseColor 0 0 1 }
18:       }
19:       geometry Sphere { radius 2 }
20:     }
21:   ]
22: }

```

Sample7改

```

1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     DEF BlueSphere Shape {
5:       appearance Appearance {
6:         material Material { diffuseColor 0 0 1 }
7:       }
8:       geometry Sphere { radius 2 }
9:     }
10:   ]
11: }
12: Transform {
13:   translation 0 5 0
14:   children [
15:     USE BlueSphere
16:   ]
17: }
18: }

```

4行目を見ると

50 DEF BlueSphere Shape

【0050】このサンプルは6つの面で立方体を表している。ポリゴンで表すためにはまず、頂点となる座標を決める(7行目乃至16行目)。この座標は上から0,1,2,と番号が振られる。即ち、10行目の「1 0 1」は「2」番の座標ということになる。次に何番の座標と何番の座標

で面を構成するかを決める(18行目乃至25行目)。19行目の「0, 1, 2, 3, -1」は「0番、1番、2番、3番の座標で面を作る」ことを表している。  
 【0051】9. ノードに名前を付ける

Sample7

```

1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     Shape {
5:       appearance Appearance {
6:         material Material { diffuseColor 0 0 1 }
7:       }
8:       geometry Sphere { radius 2 }
9:     }
10:   ]
11: }
12: Transform {
13:   translation 0 5 0
14:   children [
15:     Shape {
16:       appearance Appearance {
17:         material Material { diffuseColor 0 0 1 }
18:       }
19:       geometry Sphere { radius 2 }
20:     }
21:   ]
22: }

```

【0052】12行目以降は13行目の座標移動の部分を除いて11行目以前と全く同じである。1度定義されている「半径2で青い球」を再び同じように記述するのははっ ※30

Sample7改

```

1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     DEF BlueSphere Shape {
5:       appearance Appearance {
6:         material Material { diffuseColor 0 0 1 }
7:       }
8:       geometry Sphere { radius 2 }
9:     }
10:   ]
11: }
12: Transform {
13:   translation 0 5 0
14:   children [
15:     USE BlueSphere
16:   ]
17: }
18: }

```

4行目を見ると

50 DEF BlueSphere Shape

17

となっている。これは「Shape(……)」を「BlueSphere」という名前にします。」という意味で、これ以降 USE BlueSphere

と書くだけでShape(……)と同じ内容を表す。

【0054】10. WRLファイルの読み込み

大きなVRMLデータをつくるときに1つのファイルにすべての記述を行うのはあまりいい方法ではない。部品ごとに分けて必要に応じて呼び出せるようにしておく方が便利である。このような場合、Inlineノードを使う。例えば1.でつくったSample1.wrlを読み込んで表示する。

Sample8

```
1: #VRML V2.0 utf8
2: Inline {
3:   url [
4:     sample1.wrl
5:   ]
6: }
```

【0055】11. リンクを張る

オブジェクトにリンクを張り、オブジェクトをクリックしたときに別のページにジャンプさせることもできる。

Sample9

```
1: #VRML V2.0 utf8
2: Anchor {
3:   children [
4:     Shape {
5:       geometry Sphere { radius 2 }
6:     }
7:   ]
8:   url "test.html"
9:   description "LINK to test.html"
10: }
```

【0056】リンクを張るオブジェクトを子ノードにし、Anchorノードで括る。AnchorノードはGroupノードの1つである。リンク先のファイルをurlフィールドに書く。descriptionフィールドに書いた文字列はマウスがインタがオブジェクトに触れている間表示される。

Sample11

```
1: #VRML V2.0 utf8
2: NavigationInfo {
3:   headlight FALSE
4: }
5: Transform {
6:   children [
7:     Shape {
8:       appearance Appearance {
9:         material Material { diffuseColor 1 0 0 }
10:       }
11:       geometry Sphere {}
12:     }
13:   ]
```

18

【0057】12. ライトの設定

VRML2.0ではワールドに光を設定するためにDirectional Light (平行光)、PointLight (点光源)、SpotLight (スポットライト) 3つのノードが用意されている。ここではPointLightノードを例に挙げて説明する。光の当たり方がわかりやすいように3行目でオブジェクトを回転させている。

Sample10

```
1: #VRML V2.0 utf8
2: Transform {
3:   rotation 1 0 0 75
4:   children [
5:     PointLight {
6:       location 0 0 2 #オブジェクトの前
7:     }
8:     Shape {
9:       geometry Cylinder {}
10:     }
11:   ]
12: }
```

【0058】6行目の光源の位置によってオブジェクトの見え方が異なる。このサンプルではオブジェクトの前に光源を置いている。

【0059】13. ワールド環境(1)

これまでは主にオブジェクトの作成に関する解説であったが、今回はオブジェクト作成以外のノードの使い方について触れる。ワールド環境などと勝手に命名しているが、こういう言い方が一般的かどうかは分からない。まず、ブラウザでデフォルトで用意されている Headlight Option で変更できるが、ファイル内に書くことによっても設定できる。

19

14: }

【0060】このサンプルでブラウザの Option を見る。Headlight のチェックがはずれている。このサンプルでは今までのサンプルに比べて極端に暗くなっているのが分かる。Headlight は常にユーザの見ている方向に照らされるライトで、これがないとこのように見える。新たに加えたのは 2 行目から 4 行目の NavigationInfo ノードである。このノードの headlight フィールドを TRUE か FALSE にすることで Headlight を ON・OFF する。Headlight を消して任意のライトを設定すること 10 で効果的に明るさを設定できる。

Sample12

```
1: #VRML V2.0 utf8
2: NavigationInfo {
3:   type EXAMINE
4: }
5: Transform {
6:   children [
7:     Shape {
8:       appearance Appearance {
9:         material Material { diffuseColor 1 0 0 }
10:       }
11:       geometry Box {}
12:     }
13:   ]
14: }
```

このサンプルでは EXAMINE にしている。オブジェクトをマウスでドラッグするとオブジェクトが回転する。【0063】15. シーンにタイトルを付ける HTML では <title> タグで囲むことによってタイトルを付けることができる。もしこのタグを指定しないとタイトルが [http://ryol.is.kochi-u.howto3.html] のよ※

Sample13

```
1: #VRML V2.0 utf8
2: WorldInfo {
3:   title "Spin Box"
4:   info ["Autor H. Kan", "http://ryol.is.kochi-u.ac.jp/"]
5: }
2: NavigationInfo {
3:   type EXAMINE
4: }
5: Transform {
6:   children [
7:     Shape {
8:       appearance Appearance {
9:         material Material { diffuseColor 1 0 0 }
10:       }
11:       geometry Box {}
12:     }
13:   ]
```

※うにパスが表示される。VRML でもこれと同じことが起こる。これまでのサンプルではタイトルを指定していないためパスが表示されている。VRML でタイトルを指定するには WorldInfo ノードを使用する。

20

【0061】14. ワールド環境(2) NavigationInfo ノードには他にも幾つかのフィールドが用意されている。その中の type フィールドでナビゲーション方法を変更することができる。デフォルトでは WALK に設定されている ナビゲーション方法であるが、他にも重力を無視して移動できる FLY、自分が動くことなくオブジェクトを動かす EXAMINE、何もコントロールできない NONE、がある。ちなみに WALK は重力の影響を受けながら移動するナビゲーション方法である。

\* 【0062】

14: }  
 【0065】2行目乃至5行目までにWorldInfoノードを追加した。このサンプルでは前のサンプルに“Spin Box”というタイトルを付けた(Plugin版では表示されなかった)。尚、infoフィールドにはタイトル以外の情報を置くが、ブラウザ上には何の変化も与えない。\* 【0067】

## Sample14

```
1: #VRML V2.0 utf8
2: Viewpoint { position x y z }
3: Transform {
4:   children [
5:     Shape {
6:       appearance Appearance {
7:         material Material { diffuseColor 1 0 0 }
8:       }
9:       geometry Sphere {}
10:     }
11:   ]
12: }
13: Transform {
14:   translation -3 0 0
15:   children [
16:     Shape {
17:       appearance Appearance {
18:         material Material { diffuseColor 0 1 0 }
19:       }
20:       geometry Sphere {}
21:     }
22:   ]
23: }
24: Transform {
25:   translation 3 0 0
26:   children [
27:     Shape {
28:       appearance Appearance {
29:         material Material { diffuseColor 0 0 1 }
30:       }
31:       geometry Sphere {}
32:     }
33:   ]
34: }
```

【0068】(0, 0, 0)に赤、(-3, 0, 0)に緑、(3, 0, 0)に青の球がある。2行目のViewpointノードのpositionフィールドに具体的な座標を指定すると、そこが最初の視点になる。ただし視線方向は常にz軸の方向である。

【0069】17.視線の方向を変える  
 sample14では視点の座標のみを変更したが、もちろん視点方向も指定できる。方向を指定するときもViewpointノードを使うが、フィールドはorientationフィールドが50

かどうかを感知するTouchSensorをオブジェクトにつけてみる。  
 【0071】

```
Sample15
1: #VRML V2.0 utf8
2: Transform {
3:   children [
4:     DEF TS TouchSensor {}
5:     Shape {
6:       geometry Box {}
7:     }
8:   ]
9: }
```

【0072】これまでのサンプルと異なるのは行目だけである。ここではTSという名前を付けたTouchSensorをBoxにつけている。Boxにマウスカーソルが触れると(あなたが使っているブラウザがCommunity Place\*)

## Sample16

```
1: #VRML V2.0 utf8
2: DEF OBJ Transform {
3:   children [
4:     Shape {
5:       geometry Box { size 2 3 1 }
6:     }
7:   ]
8: }
9: DEF TS TimeSensor {
10:   cycleInterval 1
11:   loop TRUE
12:   stopTime -1
13: }
14: DEF OI OrientationInterpolator {
15:   key [0, 0.125, 0.25, 0.375, 0.5,
16:        0.625, 0.75, 0.875, 1,]
17:   keyValue [0 1 0 0, 0 1 0 0.785, 0 1 0 1.57,
18:            0 1 0 2.355, 0 1 0 3.14, 0 1 0 -2.355,
19:            0 1 0 -1.57, 0 1 0 -0.785, 0 1 0 0]
20: }
21: ROUTE TS.fraction_changed TO OI.set_fraction
22: ROUTE OI.value_changed TO OBJ.set_rotation
```

【0075】まず、幾つかのノードに名前を付けておく。2,9,14行目にそれぞれ、OBJ,TS,OIと定義している。これは後述するがイベントの受け渡しをするときに必要になる。9行目乃至13行目を見ます。TouchSensorは時間の経過を感知するノードで、時間の経過に伴って一定の間隔でイベントを生成することができ、loopフィールドはTRUEかFALSEをとるSfBoolフィールドで、TRUEならstopTimeになるまで続く。ここではstopTime -1とstartTime (デフォルトは0)とstartTimeより小さな値になっているので永久に継続されるこ

【0076】14行目乃至20行目までがOrientationInterpolatorノードである。InterpolatorノードはすべてkeyとkeyValueの2つのフィールドを持っている。keyはアニメーション時間の間隔を0から1の間で設定する。keyValueはkeyで設定した間隔に具体的なフィールド値(ここではdefRotation)を設定する。ここではアニメーション時間の間隔を9等分して、y軸を中心に回転角を設定している。

\* Browserならば)手の形に変わるはずである(他のブラウザの中にはカーソルが変わらないものもある)。ちなみにクリックしても何も起きない。

【0073】19.動きをつける(1)  
 VRML2.0がVRML1.0と大きく異なる点はシーンに動きをつけるという点である。動きをつけるにはJavaやVRMLScript(Javascript)などのスクリプトを用いる方法と、Interpolatorノードを用いる方法がある。まずはInterpolatorノードを用いる方法から見ていく。Interpolatorとは、「差し込む」とか「内挿する」といった意味である。Interpolatorノードの数値、位置、3D座標、方向、法線、色の値を変えることによってシーンに動きをつけることができる。ここでは方向を内挿するOrientationInterpolatorノードを用いてオブジェクトを回転させてみる。  
 【0074】

\* 【0066】16.視点の位置を変える

デフォルトでは視点は最初z軸上のどこか(オブジェクトの配置によって異なる)にいる。ここでは最初の視点位置を任意の場所に変更できるようにする。

【0077】しかしこれだけではシーンに動きをつけることはできない。TimeSensor ノードの生成したイベントを OrientationInterpolator ノードに渡してやる必要がある。21, 22行目を見ます。この ROUTE というキーワードで始まる行で イベントを受け渡しを行う。TimeSensor TS が起動すると fraction\_changed がイベントアクトされる。fraction\_changed がイベントアクトされると、OrientationInterpolator OI の set\_fraction イベントにインラインされる。ここまでが21 行目の ROUTE の \* Sample17

```
1: #VRML V2.0 utf8
2: DEF OBJ Transform {
3:   children [
4:     DEF TS TouchSensor {}
5:     Shape {
6:       appearance Appearance {
7:         material Material { diffuseColor 1 0 0 }
8:       }
9:       geometry Box {}
10:    ]
11:  }
12: }
13: DEF TIS TimeSensor { cycleInterval 5 }
14: DEF PI PositionInterpolator {
15:   key [0, 0.2, 0.4, 0.6, 0.8, 1.0]
16:   keyValue[0 0 0, 0 0 -2, 0 2 -4, 0 4 -6, 2 4 -6, 4 2 -6]
17: }
```

```
18: ROUTE TS.touchTime TO TIS.set_startTime
19: ROUTE TIS.fraction_changed TO PI.set_fraction
20: ROUTE PI.value_changed TO OBJ.set_translation
```

【0079】今度は、OBJ という名前の赤い立方体に TouchSensor TS をつけている。オブジェクトがクリックされると、TouchSensor TS の touchTime が TimeSensor TISの startTime にイベントアクトされる。サンプルのソースには TimeSensor ノードの中に startTime フォールドは書いてないが、デフォルトで startTime 0 となっている。あとは前のサンプルと同じである。これらの ROUTE によってオブジェクトが移動する。

【0080】ここで、第1世代のVRML 1.0 に対し、第2世代のVRML 2.0 で新たに加わった機能であるVRML 仮想空間内における自発的な動き (Behavior) を実現するためのメカニズムについて概説する。

【0081】VRML 2.0 においては、3次元仮想空間内に配置されたオブジェクトに対する操作に伴って生じるイベントや、予め設定された時間が経過した時点で生じるタイマイベントに応じて、オブジェクトの自発的な動きを実現することができる。このBehaviorのメカニズムは、センサー、ルーティング、スクリプトの3つの要素の連携動作によって実現される。

【0082】すなわち、

50 された領域内にViewPoint(ユーザーの視点)が侵入した場

合にイベントが発生するProximitySensorや、与えられた時刻となったり、予め指定された時間間隔が経過する毎に発生するTimeSensorなどが定義されている。

【0084】もう少し詳しく、Behaviorの仕組みを説明する。先に述べたように、Behaviorの仕組みは、センサー、イベント、ルーティングおよびスクリプトから構成されている。

【0085】センサーは、2つの機能に分けられる。

- ・ユーザーの操作を検知するタイプ
- ・システムの変化を検知するタイプ
- 【0086】ユーザーの操作を検知するタイプのセンサーは、3次元仮想空間内に配置された物体などに関係付けられたソフトウェアによるスイッチとなる。システムの变化を感じ取るタイプのセンサーは、作動させる時間を事前に設定しておいたタイマーが作動することになる。センサーの働きは、これらの外部イベントを検出して、それをVRML内部のイベントに変換することである。

【0087】イベントは、VRMLの内部の関連するノード間で、情報を伝えるためのデータのことを指す。実際には、VRMLファイルの中に記述されたフィールド値の変化がイベントとして伝達される。

【0088】ルーティングは、センサー機能で検出されたイベントをどのノードに伝えるかを指定するための仕組みで、イベントによる情報の伝達経路を指定することになる。

【0089】スクリプトはイベントの入出力で、入力されたイベントから何らかの計算を行い、その結果をイベントとして出力することができるものである。スクリプトは特定の言語のみに制限されるのではなく、現段階では、インターネット上で注目されているJavaやJavaScript、通常のシステムでよく利用されているC言語、UNIXでよく利用されているPerlやPython、マイクロソフト社の提供しているVisual Basic言語などが対応している。このようにVRML 2.0 では特定のスクリプト言語に依存しない仕様になっている (VRML 2.0仕様検討中、一時特定の言語仕様としてVRMLScriptが採用されようとしたこともあるが、この構想は取り止めとなった)。

【0090】次に、Behaviorの処理手段に関して、図1を参照して説明する。Behaviorの処理を図式化すると、図1のようになる。以下、各部に分けて、処理信号の流れを説明する。

【0091】①センサーノード

前に述べたように、センサーノードには、大きく分類して2系統がある。ユーザーの操作を検知するタイプと、システムの変化を検知するタイプのセンサーである。

【0092】前者のセンサーには、3次元物体やその平面をマウスをクリックしたり、通過したりしたときに感知するTouchSensorやPlaneSensorなどのセンサーノードがあり、後者のセンサーとしては、設定した時間が来るとイベントが発生させる仕組みになっているTimeSensorが

用意されている。

【0093】この図1の例では、球にTouchSensorが付けられているものとすると、このとき、ユーザーが球をマウスをクリックすると、このイベントをTouchSensorが検出する。このイベントは、TouchSensorのEventOutフィールドのフィールド値が変化することで検出される。通常1回のマウスクリックで2つのイベントが発生する。それは、マウスボタンが押されたタイミングと、離されたタイミングである。

10 【0094】次に、このイベントは、ルーティング記述部分によって、ルーティングされることになる。

【0095】②ルーティング

このイベントのルーティングを指定するのが、図2に示すような「Route」である。

【0096】センサー記述部分で発生したイベントがRouteのEventOutフィールドに伝達され、さらに、次に述べるスクリプトノードに伝えられることにより、外部ファイルにイベントが渡され、Behavior機能が実行される。

20 【0097】③スクリプトノード

このノードは、VRMLファイルと外部のスクリプトとを連動させる仲介をするノードである。スクリプトノードの記述書式に従って、どのような言語で記述されているのかとか、ファイル名を指定したり、外部スクリプトファイルとの間でイベントのやり取りをするために、eventInフィールドとeventOutフィールドを規定する。このとき、使用できるスクリプトファイルは、Java、JavaScript、C言語、Perl、およびVisual Basic言語など多岐に渡る。

【0098】実際の処理手段としては、ルーティングされたイベントがスクリプトノードに記述されているスクリプトファイルに処理が伝達され、外部のスクリプトファイルが実行される。外部のスクリプトファイルは、その中で定義されているeventIn (イベントの入力) で受け取り、そのファイル内に記述された処理を実行する。その処理が終了した後、eventOut (イベントの出力) から結果をVRMLファイルのルーティングへ返す。VRMLファイルは、その返された結果を実行して、一連のBehavior処理は終了する。

【0099】このような、センサー、ルーティング、スクリプトの連携動作によって実現されるBehaviorのメカニズムを利用することによって、例えば、3次元仮想空間内に配置されたスイッチを模したオブジェクトにマウスでクリックすることによって、同じ空間内に配置された他のオブジェクトの外観 (形状、姿勢、サイズ、色等) であるとか、挙動シーケンスなどを動的に変更する事ができる。

10100 このBehaviorのメカニズムの詳細な説明については、<http://webpage.sgi.com/moving-worlds/sp/ec/part1/concepts.html> および、その日本語版であるht



tp://www.webcity.co.jp/info/andoh/VRLM/vrm12.0/spe  
c-jp/part1/concepts.htmlで公開されている、August  
4, 1996における The Virtual Reality Modeling Lan-  
guage Version 2.0, ISO/IEC CD 14772の仕様書、4. 概  
念の節に開示されている。この節には、VRLM仕様書  
利用するにあたりキーとなる概念が記述されている。ノ  
ードをシーングラフに結合する方法、ノードがイベン  
トを生成し受け取った方法、プロダタイプによる  
ノードタイプの作成方法、VRLMにノードタイプを追加  
して外部から使用できるようにエクスポートする方法、  
VRLMファイルにプログラムとして動作するスクリプトを  
組み込む方法など、様々なノードに関する一般的な項目  
が記載されている。

【0101】次に、このようなVRLM 2. 0の自律的な動  
き(Behavior)を実現するメカニズムを応用して、共有仮  
想空間内に、仮想生命オブジェクトを誕生させ、ユーザ  
の操作や時間経過に伴う所定のイベントの発生に応じて  
変化する成長パラメータ(外観的成長又は内面的成長  
(個性))をサーバで管理し、このサーバから転送され  
た成長パラメータに基づいて、仮想生命オブジェクトの  
外観(形状、姿勢、サイズ、色等)又は挙動シナエンス  
の内の何れか一方もしくは双方を動的に変更するための  
スクリプトプログラムを解釈、実行することで、成長パ  
ラメータに応じた仮想生命オブジェクトを表示する技術  
について、以下に詳述する。

【0102】図3は本発明の一実施の形態の全体のシス  
テム構成図である。

【0103】図3において、1, 2, 3は、VRLMプラウ  
ザ及びVRLMプラウザがインストールされ、これらが動作  
しているクライアントPC(パーソナルコンピュータ)  
であり、IP(インターネット接続サービスプロバイダ)  
4, 5, 6を介してインターネット7と接続されてい  
る。

【0104】インターネット7とルータ8を介して接続  
されたLAN(Local Area Network)9には、WWWサーバ1  
0、WLS(World Location Server)11、共有サーバ1  
2、AO(Application Object)サーバ13, 14、メー  
ルサーバ15、およびコミュニケーションサーバ16が  
接続されている。これらの各サーバ10乃至16には、  
ハードディスク(HDD)10a, 10b, 11a乃至16a  
が、各々設けられている。

【0105】コミュニケーションサーバ16は、公衆電  
話回線網17を介して電話機18やファクシミリ19と  
接続され、さらに、PHS(Personal Handphone System)  
サービスプロバイダ20を介してPHS端末23に無線接  
続され、ボケットベルサービスプロバイダ21を介し  
てボケットベル端末24に無線接続されている。

【0106】図4はクライアントPC1のハードウェア  
構成を示すブロック図である。

【0107】図4において、30は各部を制御するCP

【0115】図5において、番号1で示すように、最初  
に、WWWプラウザを用いて、VRLMコンテンツを提供して  
いるWebサイトのホームページを閲覧する。この例で  
は、http://pc.sony.co.jp/separi/を閲覧してい  
る。次に、番号2で示すように、クライアントPC1と  
クライアントPC2のユーザは、VRLM 2. 0fileと、VRLM  
1空間内での自律的な動き(Behavior)を実現するため  
のスク립トプログラム(Javaによる成長スク립トプ  
ログラム)とからなるVRLMコンテンツを、それぞれダウ  
ンロードする。

【0116】勿論、CD-ROMディスク3で提供される  
VRLMコンテンツをCD-ROMドライブ32で読み込んで  
良い。

【0117】次に、図6に示すように、クライアントP  
C1及びクライアントPC2は、それぞれにダウンロー  
ドされ、一旦ローカルのHDD31に格納されたVRLM 2. 0  
fileを、VRLM 2. 0プラウザであるCommunity Place Br  
owserが解釈、実行し、さらに番号3で示すように、VSC  
P(Virtual Society Server Client Protocol)に基づい  
て、WLS11に対して共有サーバ12のURLを問い合わせ  
る。このとき番号4で示すように、WLS11はHDD11a  
に格納された共有サーバ12のURLを参照して、  
クライアントPC1及びクライアントPC2に対して、  
共有サーバ12のURLを通知する。

【0118】このURLを用いて、図7に示すように、ク  
ライアントPC1とクライアントPC2が、共有サーバ  
12に接続する。その結果、番号5で示すように、この  
共有サーバ12を介して共有3Dオブジェクトの位置や動  
きなどに関する共有メッセージの送信が行われ、番号6  
で示すように、その転送が行われ、マルチユーザ環境が  
実現される。

【0119】なお、以上の接続手順の詳しい説明につい  
ては、特開平9-81781号公報を参照されたい。

【0120】次に、共有仮想空間内に存在する仮想生命  
オブジェクトの挙動を管理するAOサーバ13について  
説明する。AOサーバ13はVSPに基づいて共有サーバ  
12と仮想生命オブジェクトに関するデータのやり取り  
を行う。また、そのHDD13aには、図7に示すよう  
に、仮想生命オブジェクトの成長パラメータ管理デー  
タや挙動コマンド管理テーブルが格納されている。

【0121】図8に示すように、成長パラメータ管理テ  
ーブルの仮想生命オブジェクトに関するデータは、仮想  
生命データと、飼い主データに大別される。

【0122】仮想生命データは、1つの共有仮想空間内  
において3Dオブジェクトを一意に特定するための3Dオブ  
ジェクトID、共有仮想空間内での仮想生命オブジェク  
トの3次元座標値、飼い主によって選ばれた猿や猫などの  
生物の種類、その性別、飼い主によって付与されたニッ  
クネーム、飼い主によって初期設定された日時、すなわ  
ちその仮想生命オブジェクトの誕生日、誕生した仮想

空間に付与されているワールド名(本稿ワールド名)  
と、仮想生命オブジェクトの成長パラメータとからな  
る。

【0123】成長パラメータは、仮想生命の外観的成長  
を規定するフィジカルパラメータと、性格等が反映され  
た内面的成長を規定するためのメンタルパラメータに大  
別される。

【0124】フィジカルパラメータは、身長(単位cm)、  
体重(単位kg)、体格指数、食欲指数、健康度指数、お  
よび寿命時間(単位時間)とからなる。

【0125】メンタルパラメータは、知能指数、言語能  
力指数、社交性指数、自主性指数、活発性指数、および  
機謙指数により構成されている。

【0126】これらのパラメータは、飼い主によって最  
初に初期設定された誕生日時からの経過時間に伴って発  
生するタイムイベントと、クライアントPCからのコ  
ールメッセージや操作メッセージに伴って発生するア  
クションイベントや操作イベントに基づき、所定の成長パ  
ラメータ算出式によって算出された値に順次更新される。

【0127】図9は、クライアントPC1のCRTモニタ  
45の画面上のVRLMプラウザのメインウィンドウに隣接  
して表示されるアクシオンパネルの機能を示している。  
【0128】この図において、(Active)と表記されてい  
るAは「呼ぶボタン」であり、仮想ペントを呼んだり、  
寝ている仮想ペントを起こす際にクリックされる。

【0129】(Sleep)と表記されているBは「寝かすボタ  
ン」であり、仮想ペントを寝かす際にクリックされる。  
【0130】Cは「食事ボタン」であり、仮想ペントに  
食事を与える際にクリックされるボタンである。

【0131】Dは「ほめるボタン」であり、仮想ペント  
に笑いかけて褒めてあげると同時にクリックされるボタンで  
ある。

【0132】Eは「あそぶボタン」であり、飼い主が鬼  
になつて逃げ回る仮想ペントを追いかけ、壁にぶつか  
って逃げない仮想ペントにぶつかって追いつめて遊  
ぶ、いわゆる鬼ごっこで遊ぶ際にクリックされるボタン  
である。

【0133】Fは「しめるボタン」であり、言う事を聞  
かない仮想ペントを叱り付けて、しつけを行なう際にク  
リックされるボタンである。

【0134】Gは「きれにするボタン」であり、仮想  
ペントをブラッシングしてきれいにする際にクリックさ  
れるボタンである。

【0135】そして、例えば、図10に示すように、自  
分のクライアントPC1において、「呼ぶボタン」Aが  
クリックされ(アクションパネルが操作され)、そのコ  
ールメッセージが共有サーバ12を介してAOサーバ1  
3へ送信されると(ステップS1)、そのアクセスベ  
ントに基づいて成長パラメータ管理テーブルの成長パラ  
メータ更新処理が実行される(ステップS3)。このア

ケースイベントに基づいて、食欲指数、健康度指数、機  
体指数が、1/10から10/10まで、0.1ポイント  
づつインクリメントされる。

【0136】また、例えば、「食事ボタン」Cがクリッ  
クされ、その操作メッセージがAOサーバー13へ送信さ  
れ（ステップS2）、操作イベントが発生する毎に、成  
長パラメータの体重が増加し、それに伴って、体格指数  
が、1/10から10/10まで、0.1ポイントづつイ  
ンクリメントされる（ステップS3）。

【0137】その後、時間経過に伴うタイマーイベン  
トが発生すると、成長パラメータの体重が減少し、それに  
伴って、体格指数が0.1ポイントづつデクリメントされ  
る（ステップS3）。

【0138】例えば、この体格指数を含む成長パラメ  
ータは、更新される毎に、共有サーバー12のマルチキャス  
ト処理（ステップS5）により、元の飼い主のクライ  
アントPC1と、仮想空間を共有しているその他のクライ  
アントPC2へ転送される（ステップS4）。

【0139】クライアントPC1では、返送されてきた  
成長パラメータに基づいて、仮想ペットの成長に伴う自  
律的な挙動を制御するための処理手順が記述された成長  
スクリプトプログラムが実行され（ステップS6）、VR  
用ファイナル仮想ペットを生成するための3Dオブジェク  
トを生成する各ノードのフィールドの値が変更され（ス  
テップS7）、この変更されたフィールドの値が反映さ  
れた仮想ペットがレンダリングされ（ステップS8）、  
クライアントPC1のCRTモニタ45の画面上のVR用ア  
ラウザのインクウインドウ上に表示される。

【0140】このクライアントPC1と同じ処理が、仮  
想空間を共有しているその他のクライアントPC2にお  
いても実行され、これにより仮想ペットの成長に伴って  
変更されたフィールドの値が反映された仮想ペットがレ  
ンダリングされ、他のクライアントPC2のCRTモニタ  
画面上のVR用アラウザのメインウインドウ上にも表示さ  
れることになる。

【0141】図11及び図12に、VR用ファイナルの仮想  
ペットを生成するための3Dオブジェクトを構成する各ノ  
ードに対応したpart0乃至part45の関係（図11）と、そ  
の表示例（図12）を示す。part0が仮想ペットの頭部  
に相当し、part1が仮想ペットの胴体に相当し、part2と  
part3が仮想ペットの右腕と左腕に相当し、part4とpart  
5が仮想ペットの右足と左足に相当する。

【0142】これらの各part0乃至part5に対応する各ノ  
ードのフィールドの値を変更することで、仮想ペットの  
各部の外観（形状、姿勢、サイズ、色彩）、及び  
各部の挙動（シークエンスを動的に変更する事が出来る。  
これは全て、成長パラメータに基づく成長スクリプトプ  
ログラムの処理で実現される。すなわち、VR用2.0で定  
義されたセンサー、ルーティン、スクリプトの連携動  
作によって実現されるBehaviorのメカニズムを利用する

50

ことによって実現される。  
【0143】従って、従来の携帯電子ペットの仮想生物  
の画像表示方法のように、予め仮想生物のキャラクター  
の成長過程の各段階のビットマップ画像をROMに格納して  
おく必要はなく、例えば、仮想ペットの成長経過に応じ  
てその体格や挙動を連続的かつ動的に変化させる事が出  
来る。

【0144】図13は、仮想ペットの成長と体格指数の  
遷移に伴って仮想ペットの体格を動的に変更して表示す  
る概念図を示している。加齢に伴って、顔が大人の顔と  
なり、体格も大きくなるが、体格指数が小さいと、やせ  
た身体となり、大きいと、大きな身体となる。

【0145】図14は、仮想ペットの機体指数の遷移に  
伴って仮想ペットの表情を動的に変更して表示する概念  
図を示している。機体指数が大きいと、笑い顔となり、  
小さいと、怒った顔になる。

【0146】図15は、仮想ペットの活発性指数の遷移  
に伴って仮想ペットの各部の挙動シークエンスを動的に変  
更して表示する概念図を示している。活発性指数が小さ  
いと、足の屈伸程度の動きしかできないが、大きくな  
ると、手を振ったり、頭を振ったりすることが出来る。

【0147】図16は、仮想ペットの知能指数の遷移に  
伴って仮想ペットへ頭蓋を付加したり眼鏡を付加して表  
示する概念図を示している。

【0148】成長パラメータの知能指数は、図9に示す  
「呼ぶボタン」Aの操作に伴うアクセスイベントに基づ  
いて、0.1ポイントづつインクリメントされ、図16に  
示すように仮想ペットの外観に影響を与える。

【0149】音階指数は、図9に示す「呼ぶボタン」A  
の操作に伴うアクセスイベントやタイマーイベントに基  
づく仮想ペットの年齢に応じて0.1ポイントづつインク  
リメントされ、仮想ペットの年齢増加に応じて、デキス  
トベースでのチャットの文章自動生成処理を実行するに  
際し、そのテキストの文体に影響を与える。例えばポイ  
ントの小さい仮想ペットのチャットは、ひらがなまたは  
カタカナで行われ、ポイントの大きな仮想ペットのチャ  
ットは、漢字の入った文字で行われる。

【0150】社交性指数は、飼い主とのチャットの頻度  
に応じて0.1ポイントづつインクリメント又はデクリメ  
ントされ、頻度が多ければ外向的な振る舞い、頻度が少  
なくなれば内向的な振る舞い、となるように仮想ペット  
の振る舞いに影響を与える。社交的で明るく、積極的な  
性格の仮想ペットは、姿勢も顔色も良くなり、逆に、内  
向的で暗く、消極的な性格の仮想ペットは、姿勢も顔色  
も悪くなる。

【0151】自主性指数は、タイマーイベントに基づく  
仮想ペットの年齢増加に応じて、0.1ポイントづつイン  
クリメントされ、次第に飼い主の言う事を聞かなくなる  
などの仮想ペットの振る舞いに影響を与える。

【0152】活発性指数は、年齢、食欲指数、健康度指

数などに基づいて決定され、図15に示すように、仮想  
ペットの振る舞いに影響を与える。また、図9に示す  
「あそぶボタン」Eの操作に伴う操作イベントの発生に  
基づいて、活発性指数が0.1ポイントづつインクリメン  
トされ、次第に活発になるなどの仮想ペットの振  
る舞いに影響を与えたり、仮想ペットの体重を減らして  
体格指数をデクリメントして図13に示すようにその外  
観に動的な変更を与える。

【0153】機体指数は、図9に示す「呼ぶボタン」A  
の操作に伴うアクセスイベントやタイマーイベントに基  
づくアクセス頻度などに基づいて決定され、図14に示  
すように、仮想ペットの表情に影響を与える。

【0154】一方、図8に示す成長パラメータ管理テー  
ブルの飼い主データは、飼い主の氏名、飼い主への連絡  
手段（連絡方法）、および、その連絡先からなる。

【0155】連絡手段が0の場合、飼い主への連絡は、  
インターネット経由の電子メールによるメッセージ文  
字音声の電話機18に対して、コミュニケーションサー  
バ16でメッセージ文のテキストデータを自動読み上げ  
ツールで音声に変換して通知する。連絡手段が9の場合  
は、PMS（PMSインターネットワークセーフティラーム標準）  
のデータ伝送方式に準拠した電子メールサービスを利用  
し、メッセージ文によってPMS端末23へ通知する。連  
絡手段が3の場合、ファクシミリ19に対して文書で  
通知する。連絡手段が4の場合は、ポケットベル端末2  
4に対してメッセージ文で通知する。

【0156】このように飼い主データを管理するのは、  
後述する既存の通信インフラストラクチャを利用した例  
い主への連絡機能や、既存の通信インフラストラクチャ  
を利用した仮想ペットへの簡易的な操作機能を実現する  
為である。

【0157】以上のシステムをまとめると、次のように  
なる。すなわち、共有仮想空間内に存在する仮想生命オ  
ブジェクトの、所定イベント（ユーザの操作又は時間経  
過に伴うイベント）の発生に応じて変化して成長パラメ  
ータ（外観的成長または内面的成長（個性））をAOサ  
ーバ13で管理し、このAOサーバー13から転送された  
成長パラメータに基づいて、クライアントPC1、2  
で、仮想生命オブジェクトの外観（形状、姿勢、サイズ  
または色）又は挙動シークエンスの内の何れか一方もしくは  
両方を動的に変更するスクリプト（プログラム）を解  
釈し、成長パラメータに応じた仮想生命オブジェクトを  
表示する。

【0158】共有仮想空間内に存在する仮想生命オブジ  
ェクトの自律的な挙動を管理するAOサーバー13に、仮  
想生命オブジェクトの、所定イベント（ユーザの操作又  
は時間経過に伴うイベント）の発生に応じて変化する成  
長パラメータを管理する管理テーブルを設け、クライ  
アントからの要求または所定イベントの発生に応じて管理

テーブルから読み出した成長パラメータを、要求元また  
はその他のクライアントの内の何れか一方もしくは双方  
のクライアントへ返送する。

【0159】成長パラメータは、上述したように、仮想  
生命オブジェクトの誕生からの所定イベント（ユーザの  
操作又は時間経過に伴うイベント）の発生に基づいて算  
出される。外観的な成長度合いを示す値である。従っ  
て、例えば、まよふ歩きの赤ちゃんから立派な大人を経  
て老人になるまでの年齢に応じた仮想生物や仮想ペッ  
ト（AO）の外観上の変化が外観的成長パラメータで規定  
される。

【0160】成長パラメータは、仮想生命オブジェク  
トに対する所定イベント（ユーザの操作又は時間経過に伴  
うイベント）の発生に起因して算出される。内面的成長  
度合い（性格）を示す値でもある。例えば、社交的で明  
るく積極的な性格の仮想生命オブジェクトは姿勢も顔色  
も良くなり、逆に内向的で暗く消極的な性格の仮想生  
命オブジェクトは姿勢も顔色も悪くなるなどの性格に応じ  
た仮想生物や仮想ペット（AO）の変化が内面的成長パ  
ラメータで規定される。

【0161】内面的成長パラメータは、仮想生命オブジ  
ェクトに対するイベントの種類に応じて異なる値が算出  
され、内面的成長度合いの値が更新される。AOサーバ  
13側で仮想ペットなどの性格を管理する場合、各クラ  
イアントから伝送されて来るメッセージの種類に応じ  
て、例えば、チャットで話かけられる毎に成長パラメ  
ータの所定の指数が0.1ポイント加算され、「ほめるボタ  
ン」が押された毎に0.2ポイント加算さ  
れ、「しかるボタン」Dが押された毎に付けられる毎に  
0.2ポイント減算される等の所定の加減算の成長計  
算式に基づいて算出される。

【0162】共有仮想空間内における仮想生命オブジェ  
クトの自律的な挙動を管理するAOサーバー13に、各仮  
想生命オブジェクトの成長度合いを示す成長パラメータ管  
理テーブルを設け、各仮想生命オブジェクトの誕生日時  
を保持し、この時点からの経過時間に基づいて、各仮想  
生命オブジェクトの年齢に応じた成長パラメータを算  
し、管理テーブルを更新する。

【0163】共有仮想空間内における複数の仮想生命オ  
ブジェクトの自律的な挙動を各々独立して管理するAO  
サーバー13（単一のAOサーバー13で複数の成長パラメ  
ータ管理テーブルを管理してもよく、複数のAOサーバ  
13、14で各々管理してもよい）を設け、それぞれの  
仮想生命オブジェクトの成長度合いを示す成長パラメータ  
管理テーブルを各々別個に設けて、各仮想生命オブジェ  
クトの成長パラメータを独立して管理することができ  
る。

【0164】その他、種々の応用例が考えられる。  
【0165】例えば、共有仮想空間における仮想生命オ  
ブジェクトの自律的な挙動を管理するAOサーバ13

に、各仮想生命オブジェクトにアクセスしたクライアントのID (ユーザが設定した仮想生命オブジェクトのニックネーム等) を管理テーブル (成長管理テーブルに、そのような管理項目を追加してもよいし、独立した管理テーブルとしてもよい。以下同様) を設け、このIDのクライアントからのアクセスに応じて親密な感情表現を示すイベントを起動するようにしてもよい。このように、AOサーバ13側で親密度 (アクセス回数) とその内容 (共有仮想空間) に入るとそばに近寄ってくるベント (仮想生命) オブジェクトを実現することができる。

【0166】仮想生命オブジェクトをセットまたはリセットしたクライアントのIDを、その仮想生命オブジェクトの飼い主として、成長パラメータ管理テーブルに格納し、仮想生命オブジェクトの寿命が尽きるまで、IDを更新し、仮想生命オブジェクトの寿命が尽きた時点でIDを消去するようにしてもよい。これにより、仮想生命オブジェクトを誕生 (セット) させた飼い主に忠誠を尽くす親密型ベントを実現することができる。さらに、寿命が尽きた (リセットされた) 時点で、自動的にその仮想生命オブジェクトの子供を誕生させ、その子供に同じ飼い主のIDを飼い主として初期設定させるようにしてもよい。これにより、その仮想ベントの子孫も先祖の飼い主になつくことになる。

【0167】仮想生命オブジェクトをセットまたはリセットしたクライアントの親密度を管理する管理テーブルをAOサーバ13に設け、クライアントからのアクセス頻度に応じて、アクセス頻度の高いクライアント程、より親密な感情表現を示す挙動シナリオを起動させることができる。逆に、アクセス頻度が少ないと、親密度が低れ、アクセス頻度で親密度が変化する挙動型ベントが実現される。

【0168】共有仮想空間内でのクライアントの位置に対応して仮想生命オブジェクトの移動を制御すれば、共有仮想空間にクライアント (飼い主) が入ってきたとき、仮想生命オブジェクトがその目の前にすぐに見え、飼い主についてまわる仮想生命オブジェクトを実現することができる。

【0169】共有仮想空間内でのイベント (例えば、他のクライアントから食事を与えられたという体験) を、仮想生命オブジェクトからテキストベースでのチャットウィンドウを介してクライアントへ体験報告させるようにすることで、親密な感情表現を行うことができる。

【0170】共有仮想空間内でのイベントを、仮想生命オブジェクトから音声チャットを介してクライアントへ体験報告することで、親密な感情表現を行うようにしてよい。テキスト読み上げソフトによるテキスト/音声変換によりアナログ音声メッセージとして音声チャット機能を介して通知してもよく、また、予め何種類かの音声メッセージをサンプリングし、圧縮してディジタル音

脈を管理し、各ユーザからのアクセス履歴に基づいて、一定期間以上アクセスのないユーザに対して、例えば、「近ごろ、会いに来てくれないので、さみしい……」などのメッセージを送信することができる。

【0177】さらに、各ユーザからのアクセス履歴に基づいて、複数のメッセージ文例の中から適切なメッセージ文を選択し、この選択したメッセージを送信するようにしてもよい。ユーザからのアクセスがあった翌日には、仮想生命オブジェクトからユーザに、「昨日は、遊んでくれてありがとう。また遊ぼうね。」などのメッセージを送り、1週間アクセスがないと、「忙しいですか? たまには、遊びに来て下さい。」などのメッセージを送る。

【0178】仮想生命オブジェクトの状態遷移を、成長パラメータ管理テーブルの更新内容に基づいて検出し、複数のメッセージ文例の中から適切なメッセージ文を選択し、この選択したメッセージを送信することができる。例えば、食事が沢山与えられた翌日には、「まだ、おなかがいっぱいだよ。」などのメッセージを送り、1週間アクセスがないと、「おなか为空いて死にそうだよ。」などのメッセージを送る。1週間1歳年を取るシナリオとした場合は、毎週、誕生日が来て年を取るたびに、「私は今日で10歳になりました。成長した姿にきて下さい。」などのメッセージを送り、寿命が尽きたら少なくなく、「そろそろ遠い所へ旅立たなければならせん、最後を見とどけて来て下さい。」などのメッセージを送る。

【0179】コミュニケーション管理テーブルのうち、電子メールに関する部分のテーブルをメールサーバ15に設け、メールサーバ15のコミュニケーション管理テーブルで、各仮想生命オブジェクトの飼い主である各ユーザの電子メールアドレス (ユーザを特定するIDも兼ねる) に対してテキストによるメッセージを送信することができる。

【0180】インターネット7を経由する電子メール以外に、PIAPS (PIAPS Internet Access Forum Standard) 方式や、oDATA32方式対応PIPSで電子メールサービスを受けられるPIPS端末23や、ページャ (ポケットベル端末24) などに対して、コミュニケーションサーバ16から公衆電話回線網17を介してメッセージを通知し、それらのLCD上にメッセージを表示させるようにしてもよい。

【0181】あるいは、コミュニケーション管理テーブルで、各仮想生命オブジェクトの飼い主である各ユーザの通知先電話番号 (ユーザを特定するIDも兼ねる) を管理し、コミュニケーションサーバ16がこの電話番号に対して自動発信して、音声によるメッセージを送信することもできる。この場合、テキスト読み上げソフトによるテキスト/音声変換によりアナログ音声メッセージとして通常の電話機8や携帯電話機 (PHS端末23)

を介して通知してもよく、また、予め何種類かの音声メッセージをサンプリングし、圧縮してディジタル音声圧縮データとしてHDDに格納しておいて、複数の音声メッセージの中から適切な音声メッセージを選択的にHDDから読み出して、伸張した後、アナログ音声メッセージとして通知するようにしてもよい。

【0182】コミュニケーション管理テーブルで、各仮想生命オブジェクトの飼い主である各ユーザの通知先アクセスID電話番号 (ユーザを特定するIDも兼ねる) を管理し、コミュニケーションサーバ16がこの電話番号に対して自動発信して、ファクシミリ19にメッセージを送信してもよい。

【0183】さらに、上述した仮想ベントからのメッセージの通知を受けた電話機18等の端末を用いて、この端末から、自分の仮想ベントに対して、簡易的な操作を行なうことで、双方向のコミュニケーションを図ることが可能となる。

【0184】既存の通信インフラストラクチャを利用して、仮想ベントへの簡易的な操作機能を実現するための具体例は以下の通りである。

【0185】例えば、共有サーバ12は、ユーザからPHS端末23、電話機18などを介してユーザを特定するIDを伴って (Caller ID等) 返送されてくる操作コマンドを解釈し、対応する仮想生命オブジェクトへのメッセージとして認識し、AOサーバ13に通知して、仮想生命オブジェクトの状態遷移 (成長パラメータの更新) へ反映させることができる。

【0186】この返信されてくる操作コマンドとしては、例えば、電話機18のブジーボタが操作されることで発信されるDTMF (Dual-tone Multi-frequency) 番号を利用することができる。

【0187】この他、例えばデネットフォニックコミュニケーションズ社の音声ブラウザブラウザCallを用いて、仮想ベントに対して各種の通知を行うようにすることもできる。この音声ブラウザによれば、通常の電話機18から音声によってサーバにアクセスすることができ、また、この音声ブラウザは、テキスト読み上げ機能を有しているので、仮想ベントからのメールを音声で聴くことも可能である。さらに、テキストをファックスや電子メールで送ることもできる。

【0188】なお、共有サーバ12、AOサーバ13、14、メールサーバ15、およびコミュニケーションサーバ16は、各種の機能を分担し、全体的なシステムとして、各クライアントPCに共有仮想空間を提供するサービスを実現している。

【0189】次に、この共有仮想空間におけるチャットについて説明する。チャットには、パブリックチャットとパーソナルチャットがある。パブリックチャットにおいては、共有仮想空間内の所定の位置に位置する1人のクライアント (ユーザ) が発生したチャットの内容が、

近傍の他のクライアント(他のユーザ)へ伝送される。  
【0190】これに対して、パーソナルチャットにおいて、対象とされる相手は指定される。この指定操作は、例えば、所定の仮想生命オブジェクトをキーボード42のシフトキーを押しながらマウスのボタンをクリックすることで行われる。このように、相手が指定された場合には、そのクリック操作を行ったクライアントが現生するチャットが、その指定した仮想生命オブジェクトのユーザに対してのみ伝送される。

【0191】チャットの内容は、ボイスチャットの場合、音声信号により伝送され、テキストチャットの場合、テキストにより伝送される。ボイスチャットの場合は、マイクロボット36を介して取り込んだ音声データが、近傍または指定した仮想生命オブジェクトのクライアント(ユーザ)に伝送され、そのクライアントPCの有するスピーカ37、38から出力される。

【0192】これに対して、テキストチャットの場合は、キーボード42を操作することで入力されたテキストが、やはり近傍または指定した仮想生命オブジェクトのクライアントPCに伝送される。そして、そのテキストは、そのクライアントPCのCRTモニタ45上に表示される。

【0193】ボイスチャットとテキストチャットのいずれを行うかは、ユーザが選択することができる。

【0194】次に、クライアントPCのCRTモニタに表示される具体的な表示例について説明する。図17は、例えばクライアントPC1から共有サーバ12にアクセスした場合のCRTモニタ45上の表示例を被示している。この表示例においては、「Community Place Bureau」に属した1のメッセージが3次元共有仮想空間の画像に重畳表示されている。なお、この画面の右側には、チャットの内容を表示する領域「Chat Log:」と、各種のアクションボタンが表示されている。

【0195】ユーザは、自分が共有仮想空間内において何者しようとするペットを選択するとき、表示画面中の「Multiuser」の項目をクリックする。このとき、図18に示すように、メニューバーが表示される。ユーザは、この中から「Select Pet」を選択する。すると、図19に示すように、「View Pet」のウインドウが表示され、このウインドウに各種のペットの画像が表示される。ユーザは、このペットの中から所望のペットを選択する。なお、このとき表示されるペットの画像は、初期状態の画像、すなわち、誕生するときの画像とされる。上述したように、このペットの画像は、成長に伴って徐々に変化していく。そして、選択が完了したとき、ユーザは、「SELECT」のボタンをマウス41を操作してオンする。

【0196】このようにして、ペットの選択が完了すると、図20に示すように、ペットに関する情報を登録する「Registration」のウインドウが表示される。ユーザ

組 腕を振む  
頷 首を傾かせる  
否 首を左右に振る  
夏 フラダンスを踊る  
右腕 動作対象を右腕にする  
左腕 動作対象を左腕にする  
右足 動作対象を右足にする  
左足 動作対象を左足にする  
上 動作対象の腕や足を上に上げる  
下 動作対象の腕や足を下に下げる  
横 動作対象の腕や足を水平方向まであげる

【0201】コマンドは、このように、人間が理解できる文字と記号の組み合わせで構成されている。このコマンドを用いて、例えば飼い主が、チャットウインドウに、「ダンス: 跳立(右腕上+左腕上) \* 2」とコマンドを入力した場合(ステップS21)、AOサーバ13において、このコマンドの解析が行われる(ステップS22)。AOサーバ13は、挙動コマンド管理テーブルを参照し、以下のように解析する。

【0202】まず「ダンス:」は、ダンスコマンドが以下に記述されていることを示しており、「跳立」は、その場で跳ねることを示し、「立」は、2本足で立つことを示し、「2」でグループ化の開始を示し、「右腕」で動作対象が右腕であることを示し、「上」で右腕を上に上げることを示し、「+」で右腕を上げながら同時に行う動作が次に示されていることを示し、その動作とは「左腕上」から、左腕を上に上げることであることを示し、「2」でグループ化の終わりを示し、さらに、「\*」でグループ化された動作を2回繰り返すことが示されていると解析する。

【0203】AOサーバ13は、このように解析されたコマンドに対して仮想ペットが行う動作を規定するスク립トを、共有サーバ12に出力する(ステップS23)。共有サーバ12は、AOサーバ13により伝送を受けたスク립トを、そのマルチキャスト処理により、元の飼い主のクライアントPC1と、仮想空間を共有しているその他のクライアントPC2へ、転送する(ステップS24)。

【0204】クライアントPC1では、返送されてきた動作パラメータ(スク립ト)に基づいて、仮想ペットのダンスなどの挙動を制御するための処理手順が記述されたダンススク립トプログラムが実行され(ステップS25)、VMDファイルの仮想ペットを表現するための3Dオブジェクトを構成する各ノードのフィールド値が変更され(ステップS26)、この変更されたフィールドの値が反映された仮想ペットがレンダリングされ(ステップS27)、クライアントPC1のCRTモニタ5の画面上的VMDブラウザのメインウインドウ上に表示される。

【0205】すなわち、図21に示したコマンド例で

は、クライアントPC1のCRTモニタ45の画面上的VMDブラウザのメインウインドウ上には、既に、そして2本足で立ち、さらにその後両腕を上げる動作を2回繰り返す犬(仮想ペット)が表示される。図23は、この動作をしている犬の所定のタイミングの表示例を示している。さらにこの図においては、チャットの内容が記載されるChat Logに、既に入力され、実行されたダンスコマンドが表示されている。

【0206】以上は、ダンスコマンドについて説明したが、その他のコマンドを用意しても良い。例えば、大なりに対して一般的に行われる「お巡り」や「伏せ」といった命令は、チャット操作により、「おすわり」、「ふせ」といったように入力されるだけで、仮想ペットがその動作を行うようにする。さらに「はなもげら」、「ふんがふ」といった、特に意味の無い言葉をコマンドとして、入力させることにより、意味のない動作をさせても良い。

【0207】仮想ペットは、飼い主が入力したコマンドに対する動作を常に行うのではなく、状況によっては、無視する場合がある。例えば、成長パラメータの指標指数が低かったり(指標が悪かったり)、知能指数が低く、命令されたコマンドに対する挙動ができなかったりする場合である。

【0208】勿論、コマンドは上述したものだけに限定されるものではない。さらに、飼い主が、自分だけのコマンドを作成できるようにしても良い。また、飼い主が仮想空間上で集まり、各自の仮想ペットにダンスをさせることにより、ダンスコンテストを開いたりすることもできる。

【0209】なお、本明細書中において、上記処理を実行するコンピュータプログラムをユーザに提供するための媒体には、磁気ディスク、CD-ROMなどの情報記録媒体の他、インターネット、デジタル衛星などのネットワークによる伝送媒体も含まれる。

【0210】  
【発明の効果】 請求項1に記載の情報処理装置、請求項3に記載の情報処理方法、および請求項4に記載のコンピュータプログラムは、サーバに送信し、サーバからそのコマンドに対する仮想生命オブジェクトの挙動を制御するスク립トの伝送を受け、このスク립トに基づいて、仮想生命オブジェクトの動作を制御するようにしたので、仮想生命オブジェクトに、複雑で、細目の動作をさせることが可能となる。

【0211】請求項5に記載の情報処理装置、請求項6に記載の情報処理方法、および請求項7に記載のコンピュータプログラムは、コマンドと仮想生命オブジェクトが行う動作との対応関係が記述されたテーブル参照してコマンドを解析し、その解析結果に基づいて受領したコマンドに対応する仮想生命オブジェクトの動作の表示を制御するスク립トをクライアント装置に伝送するようにしたの

で、仮想生命オブジェクトに、模様で、独自の動作をさせることが可能な共有仮想空間を提供することができ

る。

【図面の簡単な説明】  
【図1】 センサ、イベント、ルーティング、およびスクリプトの関係を示す図である。

【図2】 ルーティングを説明する図である。

【図3】 本発明を用いた共有仮想空間提供システム構成例を示すブロック図である。

【図4】 図1のクライアントPC1の構成例を示すブロック図である。

【図5】 図3のシステムの動作を説明するディスプレイ上に表示した中間画面像の写真である。

【図6】 図3のシステムの動作を説明するディスプレイ上に表示した中間画面像の写真である。

【図7】 図3のシステムの動作を説明するディスプレイ上に表示した中間画面像の写真である。

【図8】 成長パラメータ管理テーブルの例を示す図である。

【図9】 アクションパネルを説明するディスプレイ上に表示した中間画面像の写真である。

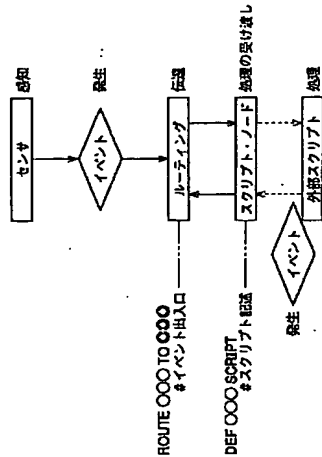
【図10】 成長パラメータの授受を説明する図である。

【図11】 3Dオブジェクトを構成する仮想生命オブジェクトのノードを説明する図である。

【図12】 図11に示したノードに対応する表示例を示すディスプレイ上の中間画面像の写真である。

【図13】 仮想生命オブジェクトの体格指数を説明する図である。

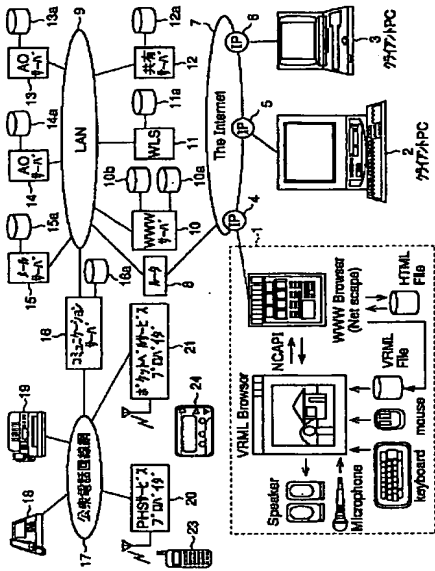
【図1】



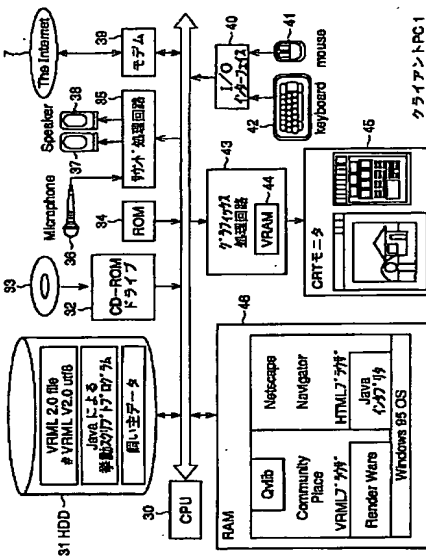
【図2】

ROUTE Node Name, event Out Name TO Node Name, event In Name  
(ルート名) (ノード名) (イベント名) (ノード名) (イベント名)

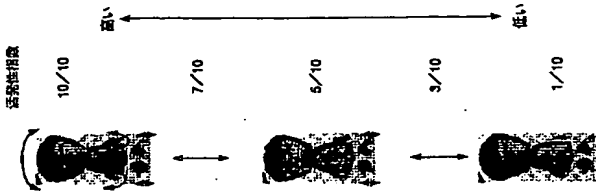
【図3】



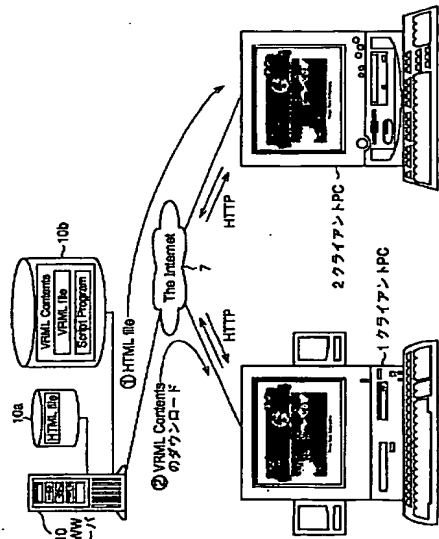
【図4】



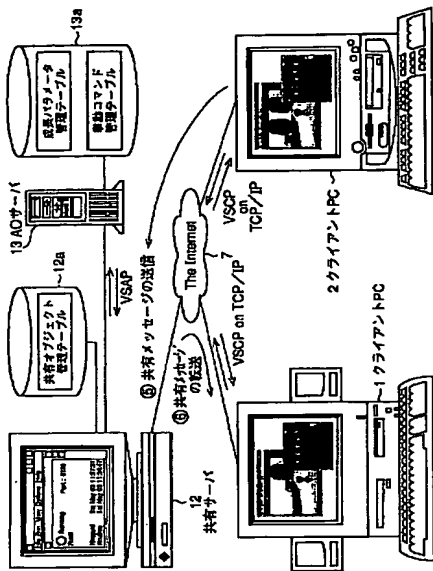
【図15】



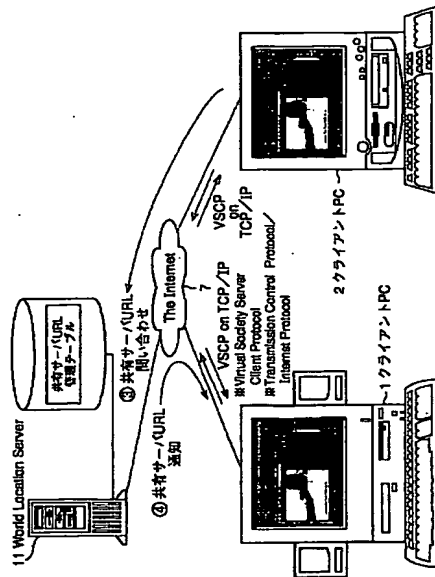
【図5】



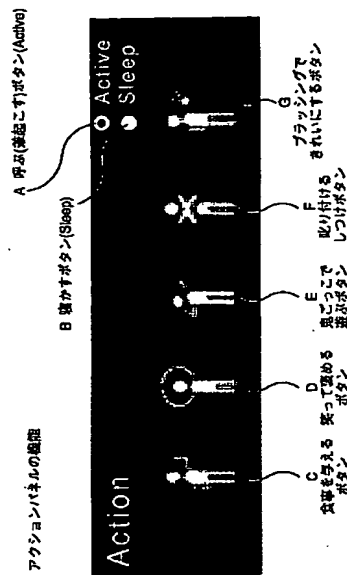
【図7】



【図6】

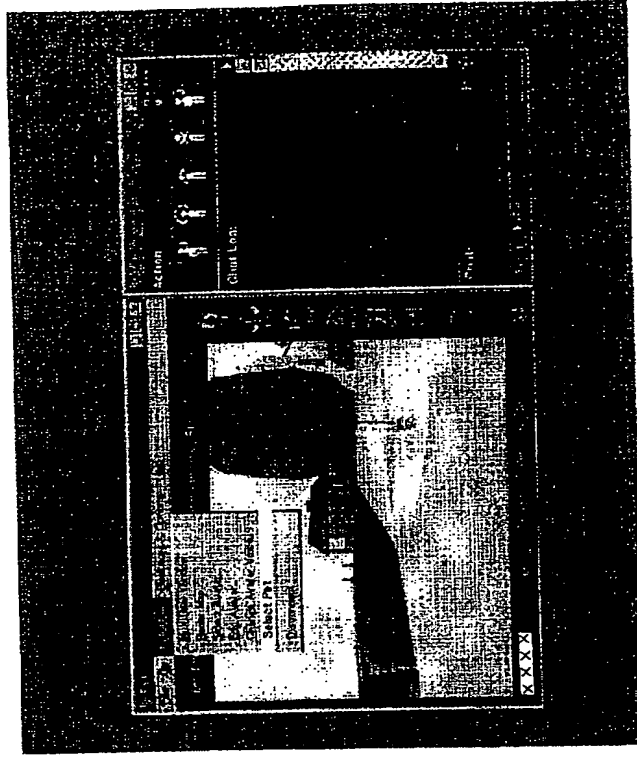


【図9】

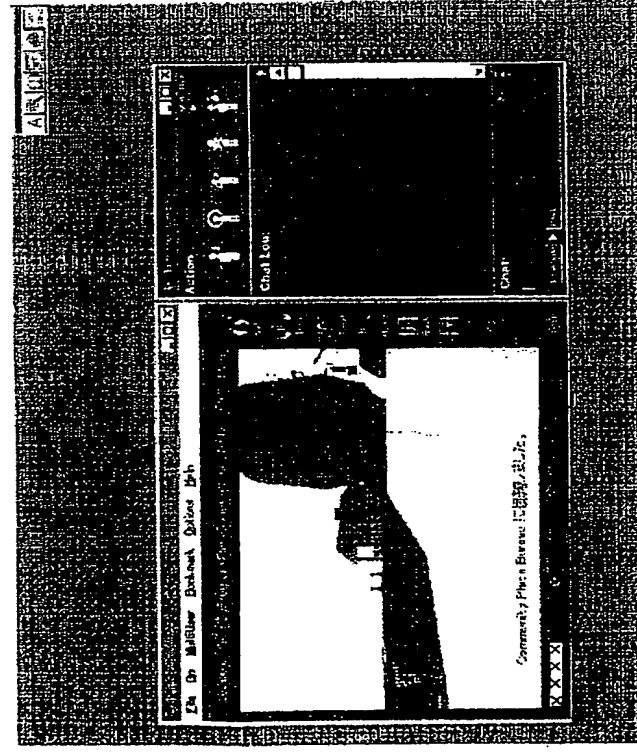




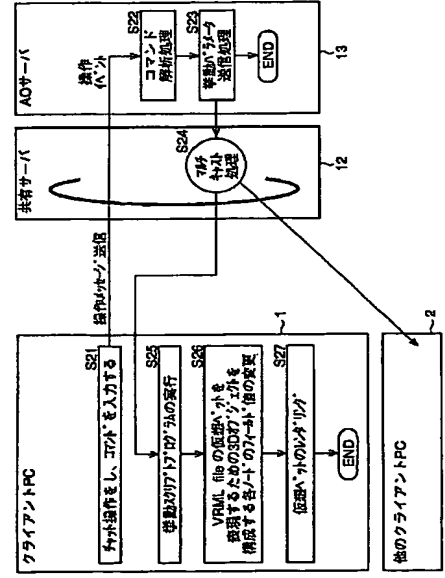
【图18】



【图17】

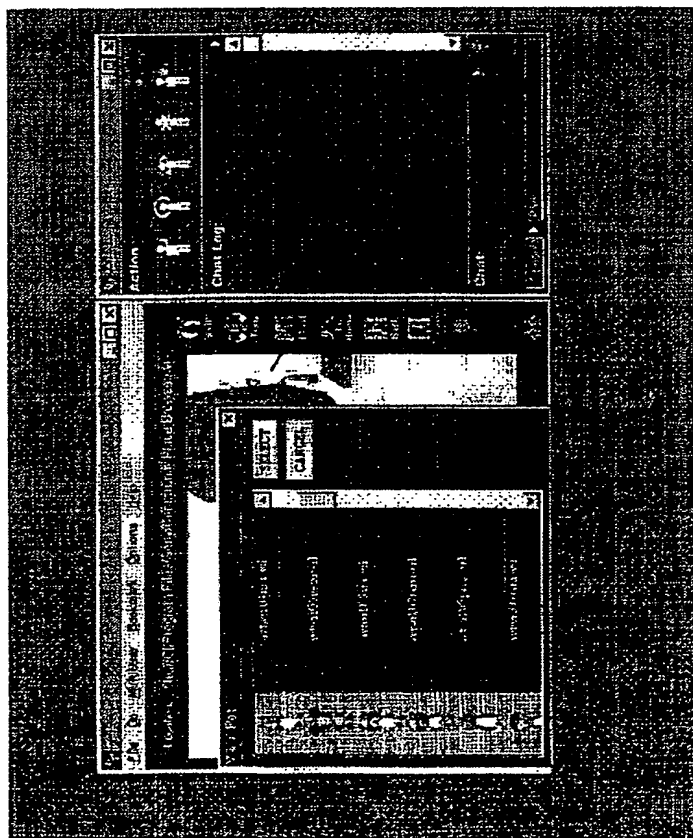


**[22]**

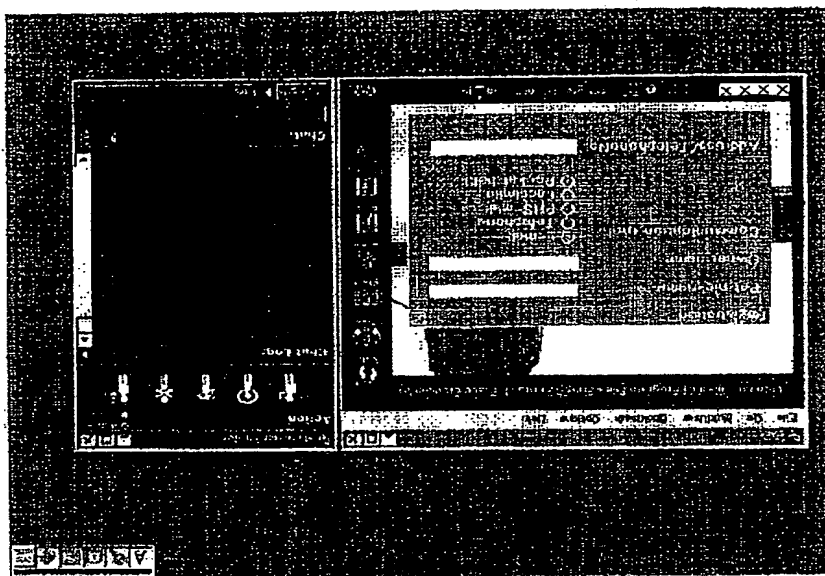




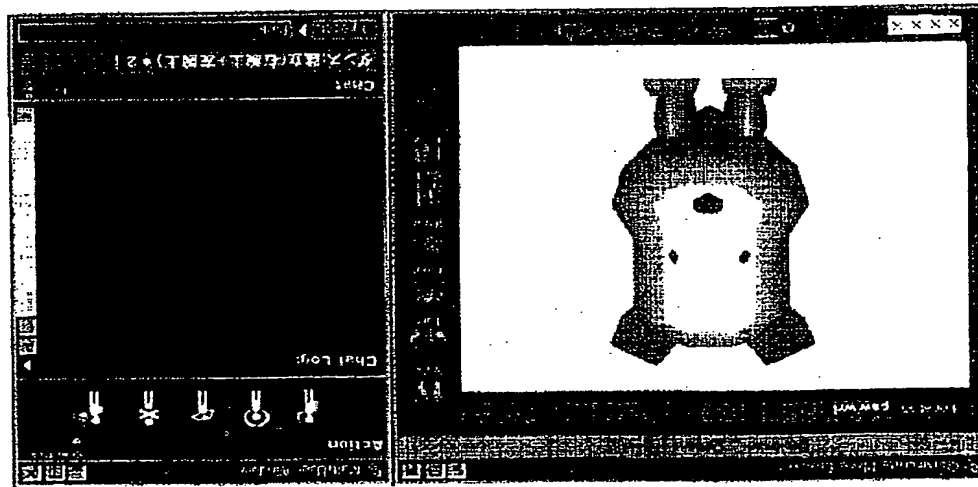
【図19】



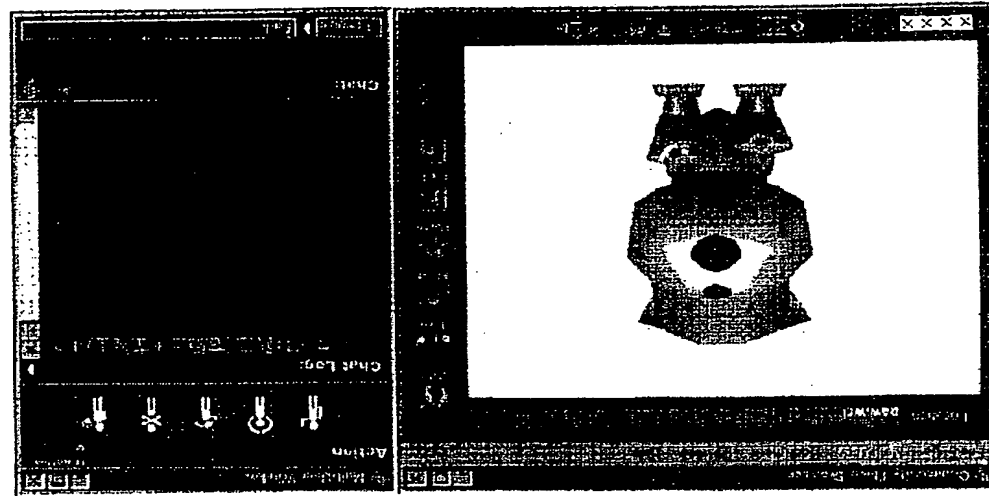
【図20】



【図21】



【図23】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**